Recursive Algorithms and Fractals

1st Akaash Parthasarathy
Honors Discrete Mathematics
Georgia Institute of Technology
Atlanta, USA
arp9@gatech.edu

2nd Rohan Phadnis
Honors Discrete Mathematics
Georgia Institute of Technology
Atlanta, USA
rphadnis6@gatech.edu

3rd Ansh Bhatti
Honors Discrete Mathematics
Georgia Institute of Technology
Atlanta, USA
abhatti33@gatech.edu

***Abstract*—This paper explores recursive algorithms and fractals in theory, analyzes common real-world examples, examines a variety of their applications, and delineates an experiment design pertaining to fractal antennae and how their geometric properties enhance signal strength and bandwidth.**

***Index Terms*—fractals, image compression, fractal antennas, Mandelbrot set, Koch snowflake**

I. Introduction

Recursion and fractals are abundant in nature. Perhaps the most ubiquitous example of these concepts are trees. Trees are composed of a large trunk from which several branches emerge. Each of these branches further gives rise to a number of smaller branches and so forth. The trunk of a tree can thus be traced to individual twigs holding leaves. We wish to consider what would happen if the branches of trees continued to give rise to smaller branches infinitely. This leads us to the definition of recursion.

Recursion is the repeated application of an algorithm or recurrence relation. One of the simplest examples of a recursive definition is that of the factorial of a number. The factorial of a number n, denoted by $n!$, is calculated as $n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$. Factorials can be defined recursively as

$$0! = 1$$

$$n! = n \cdot (n-1)!$$

In computer science applications, recursion is limited by a base case, for which the answer is known beforehand and beyond which recursion is halted. For the factorial example, the base case is $0! = 1$. For our discussion, we are going to deal with a special case of recursion called infinite recursion, that is, recursion without a base case.

A system of recursive algorithms can be used to generate fractals, geometric shapes that can be divided into smaller copies of themselves. Many fractals are self-similar, resulting in ceaseless patterns that take advantage of the fact that there are infinitely many reals. Essentially, a copy of a shape can be smaller but still exist. Besides their abundance in nature, fractals also find several important applications in mathematics. For example, Julia sets are instances of quasi-self-similar fractals while the coastline of Britain is an example of a fractal that is not self-similar. A few further common mathematical examples involving fractals are the Koch Snowflake, the Sierpinski Gasket, and the Mandelbrot Set as we will discuss in the following sections.

II. Fractals and Self-Similarity: The Koch Curve

Fractals have a number of properties that help describe their shape. The side property of a fractal represents the number of sides the base shape of a fractal has and is usually represented by S. The scale factor property of a fractal represents the scale by which each deeper appearance of the shape is dilated down by and is usually represented by N. In addition, fractals allow us to determine the "roughness" of objects. The roughness of a fractal measures its quality and variability as a geometric surface. It is calculated as $R = \frac{\log(N)}{\log(S)}$, where $\log(x)$ is the base 10 logarithm of x. Note that the formula for roughness can be generalized for all possible values of N and S. Roughness can be equated with irregularity. For example, the Earth as viewed from thousands of miles from space appears to be a perfect sphere, but the further one zooms in, the further the irregularity of the surface is explored. The concept of roughness allows for many applications to be achievable. For games, this means the development and rendering of realistic landscapes (compression and decoding of images as we discuss later). For the health sector, this means modeling of the complex human body. For cartographers, this means piecing together the irregular building blocks of land borders on a piece of paper. Although we do not discuss roughness in depth, we note that using fractals to calculate roughness finds applications in a wide variety of fields that perform modeling of complex phenomena.

Although fractals are not necessarily self-similar, many fractals exhibit self-similarity. The definition of self-similarity is as follows: "Given a real ratio parameter r, a positive integer N, and a bounded set S of points $(s_1, s_2, s_3, ..., s_E)$ defined in a Euclidean space of dimension E, then S is self-similar if it is the union of

N non-overlapping subsets, each of which is congruent to rS." More formally, given r, N, and S where $r \in \mathbb{R}$, $N \in \mathbb{Z}^+$, and $S = \{s_i | (s_i \in R^E) \wedge (1 \leq i \leq E)\}$, the set S is self-similar if $x_0 \cup ... \cup x_N = S$ where $(x_j \subset S) \wedge (x_j \cong rS) \wedge 1 \leq j \leq N$.

Here, a Euclidean space is simply a space of some positive integer dimension E where Euclidean geometry is applicable. In this definition, the congruence of sets refers to the idea that the sets coincide after a suitable affine transformation. For the purposes of this project, we elect to use a simpler version of this definition: we define self-similarity as a property of geometric shapes in which a structure is made up of nearly perfect smaller copies of itself indefinitely.

We explain this idea in terms of the Koch curve, a fractal curve discovered by the Swedish mathematician Helge van Koch. A more famous fractal derived from the Koch curve is the Koch snowflake. The Koch curve can be constructed beginning with a line segment and recursively modifying the line segment using the following procedure:

1) Divide the line segment into three line segments of equal length
2) Construct an equilateral triangle pointing outward such that the central line segment is its base
3) Remove the line segment that forms the base of the constructed equilateral triangle

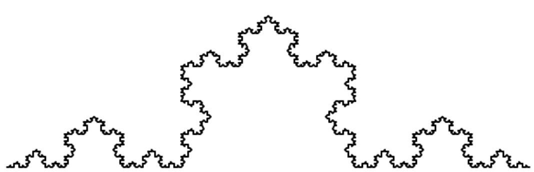


Fig. 1: Koch curve generated using Python.

The Koch curve is produced when this recursive procedure is continued to infinity. The construction of the Koch snowflake is similar and applies this procedure to the three line segments of an equilateral triangle. The illustration of the Koch curve in Fig. 1 was generated using the Turtle module in Python (see the Appendix for the code).

The length of the Koch curve diverges as the depth of recursion tends to infinity. We verify this fact as follows. Since each recursive call multiplies the number of line segments in the Koch curve by four, the number of line segments in the Koch curve after n recursive calls is given by

$$N_n = N_{n-1} \cdot 4 = 4^n$$

If the original line segment has length s, the length of each line segment in the Koch curve after n recursive calls is given by

$$S_n = \frac{S_{n-1}}{3} = \frac{s}{3^n}$$

The total length of the Koch curve after n recursive calls is then given by

$$L_n = N_n \cdot S_n = 4^n \cdot \frac{s}{3^n} = s \cdot \left(\frac{4}{3}\right)^n$$

We note that as n tends to infinity

$$\lim_{n \to \infty} L_n = \lim_{n \to \infty} s \cdot \left(\frac{4}{3}\right)^n = \infty$$

Thus, the Koch curve has infinite length. As such, it can be shown that the Koch snowflake has an infinite perimeter. This property of the Koch snowflake, and of fractals in general, is used in the construction of fractal antennae (see Section IV-B).

III. An Important Fractal: The Mandelbrot Set

We focus our discussion in this section on a particularly important fractal called the Mandelbrot set. The Mandelbrot set is a self-similar fractal (although, mathematically speaking, the boundary of the Mandelbrot set is a self-similar fractal, these terms are often used interchangeably) obtained from the following quadratic recurrence relation:

$$z_0 = 0$$
$$z_{n+1} = z_n^2 + C$$

where C is a complex number and $n \geq 0$. The set consists of all points C in the complex plane for which the orbit of the point $z = 0$ does not diverge under repeated application of the recurrence relation. In other words, the complex number C belongs to the Mandelbrot set if z_n is bounded for $n \geq 0$. For example, when $C = 1$, repeated application of the recurrence relation yields the sequence $0, 1, 2, 5, 26, 677, \ldots,$ which clearly diverges as n grows without bound. Thus, $C = 1$ does not belong to the Mandelbrot set. On the other hand, when $C = -1$, the generated sequence is $0, -1, 0, -1, 0, \ldots,$ which is bounded. Thus, $C = -1$ is a member of the Mandelbrot set.

The Mandelbrot set is a special case of a more general class of fractal sets called Julia sets. It is composed all points in the complex plane for which the corresponding Julia set is connected (in the sense that it is possible to get from each point in the set to every other point in the set without ever having to leave the set). Whether the corresponding Julia set is connected or disconnected depends entirely on the value of C chosen for the recurrence relation. The Mandelbrot set is thus often treated as a map of connected Julia sets.

The main figure in the plot of the Mandelbrot set is a circle with center $(-1, 0)$ and radius $\frac{1}{4}$ adjoined to a cardioid whose equations are:

$$4x = 2\cos t - \cos(2t)$$
$$4y = 2\sin t - \sin(2t)$$

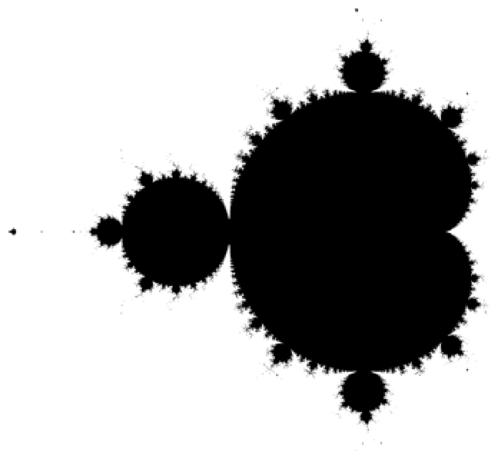
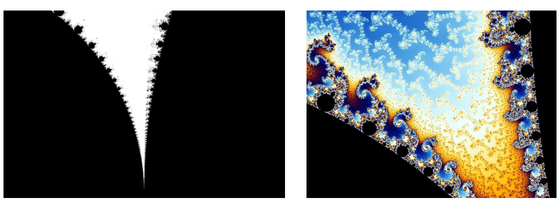
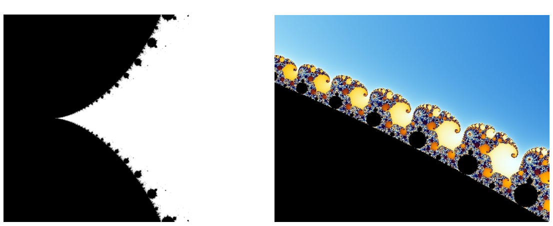


Fig. 2: Mandelbrot set generated using Python.



(a) Standard view **(b)** Zoomed-in view

Fig. 3: The spirals in the sea horse valley resemble sea horse tails.



(a) Standard view **(b)** Zoomed-in view

Fig. 4: The spirals in the elephant valley resemble elephants in a parade.

Fig. 2 is a rough representation of the Mandelbrot set generated using Python (see the Appendix for the code). The Mandelbrot set contains several interesting patterns. The region of the set centered around $-0.75 + 0.1i$ is often called the "sea horse valley" owing to the fact that the spirals when zoomed in look like sea horse tails (see Fig. 3). Similarly, the region of the set centered around $0.3 + 0i$ is called the "elephant valley" since the spirals resemble elephants marching in a parade (see Fig. 4).

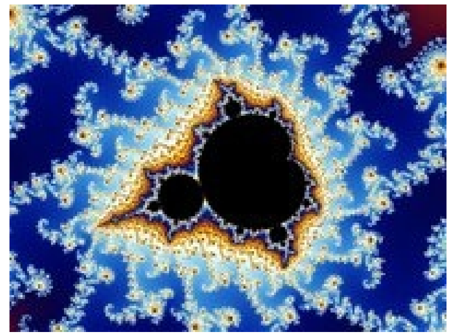


Fig. 5: A smaller version of the Mandelbrot set within the Mandelbrot set.

The sea horse valley is particularly important because zooming in on it sufficiently far enough yields a smaller copy of the Mandelbrot set as shown below. This smaller Mandelbrot set possesses its own sea horse valley, which in turn contains an even smaller copy of the Mandelbrot set (see Fig. 5). This continues indefinitely, lending the property of self-similarity to Mandelbrot sets.

Fig. 6: Quantization of shades at discrete levels of the Mandelbrot set produces equipotential curves.

An important term for the generation of computer visualizations of the Mandelbrot set is the escape radius, denoted by r_{max}. Since it is computationally intractable to check with certainty that a particular complex number C does not cause the sequence produced by the recurrence relation to diverge, we define r_{max} (instead of infinity) as the limit at which the sequence is assumed to have diverged. Often r_{max} is chosen to be 2 even though this value is arbitrary. For computer visualizations, by convention, if a Mandelbrot sequence either converges or oscillates at a value below the escape radius, the corresponding pixel is assigned the color black. On the other hand, if the Mandelbrot sequence diverges past the escape radius, the corresponding pixel is assigned a color depending on how fast the sequence diverges. If the sequence diverges slowly, the pixel is assigned a darker color, and vice versa. A Python-generated visualization of the Mandelbrot set following this scheme can be seen in Fig. 6 (see the Appendix for the code). Notice that the shades appear quantized at discrete levels. These correspond to curves called Mandelbrot set lemniscates or equipotential curves, which are often used for applications such as terrain navigation and mapping.

IV. Applications of Self-Similar Fractals

Fractals have numerous applications in fields ranging from fluidics, medicine, image processing, and electrical engineering.

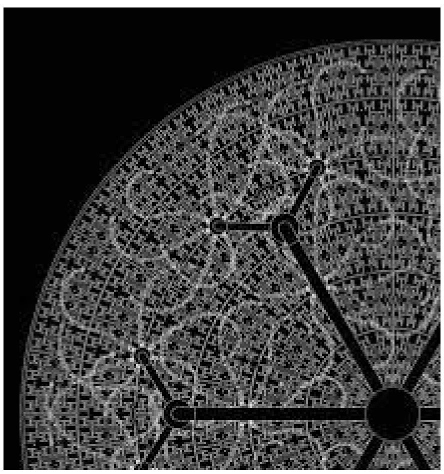


Fig. 7: Fractal apparatus used by Amalgamated Research Inc. for fluid mixing.

For example, Amalgamated Research Inc. is pioneering in fractal-based fluid mixing technology. Their devices are designed to optimize precision in a variety of mixing processing including chromatography, ion exchange, absorption, and distillation. See Fig. 7 for an image of the company's apparatus.

Fractal theory is also being used to analyze CT scan imagery for diagnosing lung disease. Given the tree-like self-similar structures of the bronchial networks in the lungs, an analysis of observed discrepancies in shape and roughness of certain branches is used to reveal potential medical problems.

Fractals are additionally used for image compression. Fractal compression is an image data compression algorithm. It is a type of lossy conversion, meaning that some data is lost in the compression process. This compression technique is used for compressing images of repetitive patterns, complex landscapes, and nature. The algorithm extrapolates fractal codes from the image data. These codes are then used to re-create the image from its compressed form. An iterated function system (IFS) is used to describe fractals mathematically using set theory and function notation. IFSs are used to render an image from its compressed encoding.

Finally, Fractal Antenna System Inc. is developing fractal-shaped antennae for mainstream communication devices such as mobile phones. According to research, these fractal antennae can capture a wider range of frequencies, transmit and receive data at faster rates, establish stronger and more reliable signals, and require fewer components to assemble.

A. Fractal Image Compression and Decoding

Images are stored as 0s and 1s on computers. They occupy significantly higher sections of memory than text data, storing per-pixel data. Researchers have worked on image compressions for the past few decades in order to store images in less memory, reduce storage cost, and be able to transmit data more efficiently. Luckily, as large as images may be, they have found to proportionally contain degrees of redundancy. Image compression is all about being efficient with image storage while making the changes negligbly noticeable to the human eye. Common methods include JPEG, which is a form of lossy compression that removes high-frequency noise. Techniques like these, however, are largely resolution-dependent. For example, a JPEG image cannot be decoded and resized to provide a larger quality.

As shown in various animations, no matter the scale of the zoom, fractals will keep multiplying and maintain the same level of roughness. The self-similar properties of fractals can be used to compress images and restore them independent of resolution by extracting the self-similarity of an image from the redundancy in it. For example, a 64x64 image can be encoded and restored to an image of 256x256, retaining a much better quality than otherwise obtained by simply resizing/zooming into the image.

Every next, similar level of a fractal can be obtained by a transformation of the existing, preceding level. Such a transformation is contractive, generating smaller versions of levels of fractals. A contractive transformation is one that for any two points x and y, the distance between $T(x)$ and $T(y)$ is smaller than the distance between x and y. This contractive transformation must be an affine transformation, one which scales, skews, rotates, stretches, or translates points. According to the Contractive Mapping Fixed Point Theorem, repeatedly applying such a transformation will eventually converge the two points x and y in a complete metric space. Similarly, a contractive transformation for images, when applied repeatedly, would converge all those images

(a)

(b)

(c)

Fig. 8: Demonstration of fractal image decoding on zoomed in monkey image. (a) Original image, (b) Zoomed-in image, (c) Decoded image.

to one fixed image. For image compression, this fixed image must be the image that needs to be compressed. What's the point of this transformation when we are mapping the original image onto itself?

Turns out, that's not what's exactly happening. We will discuss block fractal encoding, which involves taking equal sized blocks of an image and then applying the appropriate contractive transformations to each block in order to get back the original image. Essentially, fractal encoding compresses an image into an ensemble of transformations that can help decode back to the original image when applied to **any random image**. Note that these transformations aim to depict self-similarity in an image, so to form them, we need to compare parts of the image to a version of itself. One way of doing it is comparing the blocks of the original image (range blocks) to blocks of the half-sized version of that image (domain blocks), with the blocks all still being the same size. For every range block $R_{k,l}$, we want to find a domain block $D_{i,j}$ and an affine transformation τ such that the distance between $\tau(D_{i,j})$ and $R_{k,l}$ is minimized: min $\sum_{x=1}^{\mathrm{H}} \sum_{y=1}^{\mathrm{W}} (R_{k,l} - \tau(D_{i,j}))^2$ where W = block's width and H = block's height. Once the transformation and the domain block have been found to result in the best approximation for that range block, the transformation's coefficients and domain block's coordinates are stored for that range block.

(a)

(b)

(c)

Fig. 9: Demonstration of fractal image decoding on Lena image corrupted by 10% salt and pepper noise. (a) Original image, (b) Corrupt image, (c) Decoded image.

Fractal image compression uses the geometric properties of an image, so the compressed image can even be decoded to a larger size than the original image itself, with minimal pixelization and greater emphasis on detail. Essentially, the picture looks good when generated with transformations even for an increased size. Examples of this process in action can be seen in Fig. 8 and Fig. 9.

B. Fractal Antennae and the Koch Curve

It is possible to construct antennae for communication tasks using self-similar fractal shapes. Research has shown that these fractal antennae can capture a wider range of frequencies, transmit and receive data at faster rates, and establish stronger and more reliable signals. In addition, they require fewer components to assemble.

To understand the science behind fractal antennas, we must first understand how antennas work. Antennas are utilized in transmitters and receivers. In a transmitter, current is passed through a wire to generate an electromagnetic field around it. The direction of current flow can be alternated to create a certain frequency. On the receiving end, current is induced in the receiving antenna at that frequency. Electrical components in the

(a) Iteration 0 **(b)** Iteration 1 **(c)** Iteration 2

Fig. 10: Koch Snowflake Monopole Antennae constructed for the experiment.

receiver are used to detect these signals and interpret the communication.

As the prevalence of reliable communication in constrained environments increases, the necessity of smaller communication infrastructure that works at a wide range of frequencies also increases. Research has shown that fractal-shaped antennas can solve this problem.

A resonant frequency of an oscillator is the natural frequency at which it oscillates. This allows an oscillator to achieve its highest amplitude, thereby strengthening the signal and decreasing noise. Since resonance is reached upon reaching the resonant frequency or its higher multiples, in radio communication, it is preferred to have lower resonant frequencies. According to the study, "On the resonant properties of the Koch fractal and other wire monopole antennas," increasing the number of iterations in a fractal antennae such as the Koch Snowflake, Rectangular Meander Line, Triangular Meander Line, and Normal Mode Helix, decreases the resonant frequency. Additionally, his research shows that increasing iterations also decreases the resistance radiation (a measure of dissipated energy) and the standing wave radio bandwidth (a measure of signal error), all while only needing minimal increments in wire usage.

The reliability of the signal is also increased since increasing recursive depth generally increases a fractal's roughness, thereby increasing its perimeter. An increase in perimeter increases the amount of material that can receive and transmit radiation, allowing the antenna to be more receptive to electromagnetic flux.

C. Experiment Design: Fractal and Traditional Antennae

As a proof of concept to test the claim that fractal antennae are better than traditional monopole antennae, we designed and conducted an experiment using low-cost electrical components.

1) Hypothesis: Our hypothesis is that as the number of iterations of the Koch Snowflake in the antennae are increased, the signal's reliability also increases.

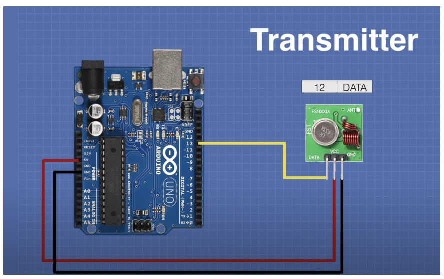
2) Setup: Table I lists the bill of materials we used for the project. For this experiment, we compared 3 different setups. They are as follows:

- Iteration 0 Koch Snowflake Monopole Antenna to approximate currently used monopole antennae
- Iteration 1 Koch Snowflake Monopole Antenna
- Iteration 2 Koch Snowflake Monopole Antenna

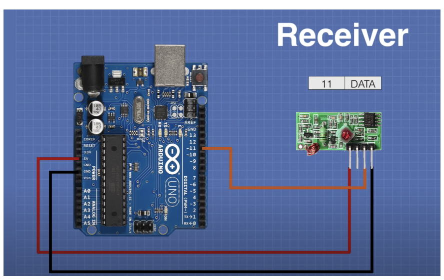
For each Antenna, we designed a basic "tuning fork" to allow branches to form. We folded 17.3cm of tin wire into Koch Curves as 17.3cm is the optimal length for the 433mHz frequency. This length is controlled at a constant for all setups. 4 Koch Curves were designed and soldered to each "tuning fork." Each of these antennae were soldered to an RF receiver. See Fig. 10 for images of the antennae.

TABLE I: Bill of Materials

Item	Quantity
Arduino Uno Board	2x
Arduino Uno USB Cable	2x
433mHz Radio Frequency Transmitter	1x
433mHz Radio Frequency Receiver	3x
Tin Wire	3.2m
Wood Planks	3x
Electrical Tape	1 roll
25ft Measuring Tape	1x



(a) Transmitter circuit



(b) Receiver circuit

Fig. 11: Circuits designed for the experiment.

To connect the transmitter and receiver to the Arduino, we followed the circuits in Fig. 11. We ran a program on the transmitter end to send the message consisting of an integer value and the text "Hello!" at 1 second intervals. This integer value was initiated to 0 and incremented by 1 each time a message was sent. We programmed the receiver end to interpret these messages and print them to the serial monitor. See the appendix for the code for both of these programs.

3) Experimental Procedure: As it is with RF communication, the signal weakens with distance. A stronger antenna expands the communication range. We wanted to test if increasing the iterations increases the maximum communication distance of the antennae. We followed the below outlined procedure thrice for each antenna and averaged the values.

TABLE II: Experimental Data

Iterations	Maximum Range
0	18.379m
1	25.512m
2	36.064m

We initiated the communication system at a separation distance of 0m. Then, we displaced the transmitter from the receiver until the communication became unreliable (as measured by how many messages were not being received). We increased the separation distance until the transmitter and receiver were not in each others' range. Our threshold for out-of-range in this experiment was whether a message was not received from a particular distance for a period of 30 seconds. We measured and recorded this maximum separation distance.

After conducting the experiment, we found that increasing the iterations did, indeed, increase the communication range.

4) Results: As can be seen in Table II, the number of iterations increases the range of the RF communications. This supports the theory and past experimental research into fractal antennae. To reiterate, the self-similar nature of the Koch Snowflake and surface area is optimized to allow picking up a wider range of frequencies, enabling the modules to communicate for a longer distance. That said, there may have been sources of error in the experiment such as inaccuracies with measuring, circuit hardware problems, faulty parts, and random noise.

5) Conclusion: In spite of the numerous benefits of fractal antennae as shown by past research, simulations, and experiments, most antennae used in conventional applications are still linear, parabolic, or of other forms. The explanation for this is likely convenience. Fractal antennae are very tedious to integrate into the industrial process and manufacture at scale. Their intricate shapes will be an inevitable bottleneck in practice.

Appendix
Code

The codebase for the project can be found at the following link: https://github.com/akaashrp/recursive_algorithms_and_fractals.

The code used to generate Fig. 1 was written using the Turtle module in Python. It depicts 5 recursive calls of the procedure for generating the Koch curve. At each recursive call, the length of the line segment is reduced to one-third the original length, and the turtle is oriented in the appropriate direction.

The code used to generate Fig. 2, Fig. 3(a), and Fig. 4(a) was written using the NumPy and Matplotlib libraries in Python. It defines a complex matrix and colors in the pixels corresponding to entries that do not diverge under the recursive procedure.

The code used to generate Fig. 6 was written using the PIL library in Python. It defines a maximum number of iterations and an escape count to determine how long it takes different input values to diverge.

The code for the transmitter and receiver in Section IV-C was written using the RadioHead library for the Arduino programming language.

ACKNOWLEDGMENT

Firstly, we would like to thank Professor Gerandy Brito for equipping us with knowledge of discrete mathematics, as well as giving us the opportunity to undertake this project. Secondly, we would like to thank Mr. Mohanty for his feedback over the course of our project. Thirdly, we would like to thank the authors of all references we have cited in the paper for contributing to the ever-growing repository of scientific knowledge. Fourthly, we thank the Hive Makerspace and all involved PIs for letting us use the facility, tools, parts, and machines. Last, but not the least, we would like to thank each other for our collaboration, friendship, and the memories we made while working on this project, which we shall cherish forevermore.

REFERENCES

[1] A. Gupta, “Digital Image Processing - Open Project,” Fractal Image Compression. [Online]. [Accessed: 23-Apr-2023].
[2] B. B. Mandelbrot, The fractal geometry of nature. New York: Freeman, 1983.
[3] B. Zaczyński, “Draw the Mandelbrot set in Python,” Real Python, 25-Jan-2023. [Online]. [Accessed: 23-Apr-2023].
[4] “Barnsley’s Tree,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023].
[5] D. Roy, “Exploring Fractal Dimension.” University of Sydney, 02-Jun-2019.
[6] “Delannoy number,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023].
[7] “Elephant Valley,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023]. Fractals are typically not self-similar. YouTube, 2017.
[8] “Fractals,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023].
[9] “An Introduction to Fractal Image Compression.” Texas Instruments, Oct-1997. Javascript Julia Set Generator. [Online]. [Accessed: 23-Apr-2023].
[10] “Koch snowflake,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023].
[11] M. McGoodwin, “Julia Jewels: An Exploration of Julia Sets,” Julia Sets, Mar-2000. [Online].
[12] “Mandelbrot Set,” Wolfram MathWorld. [Online]. [Accessed: 23-Apr-2023].
[13] ManimCommunity, “Manimcommunity/Manim: A community-maintained python framework for creating mathematical animations.,” GitHub. [Online]. [Accessed: 23-Apr-2023].
[14] S. Hooks, “Interesting applications of fractals,” Medium, 06-Nov-2021. [Online]. [Accessed: 23-Apr-2023].
[15] “Unit 9 Fractal Chaos.” University of Pennsylvania, 2013.
[16] Using Inexpensive 433 MHz RF Modules with Arduino. YouTube, 2018. https://www.overleaf.com/project/640bac2c3aa60f30418b5450
[17] W. Ruckman, “HOW TO MAKE A FRACTAL ANTENNA FOR HDTV / DTV PLUS MORE ON THE CHEAP,” Ruckman.net, 2008. [Online]. [Accessed: 23-Apr-2023].
[18] W.-Y. Yi, “Visual Exploration,” The Mandelbrot Set, 2006. [Online]. [Accessed: 23-Apr-2023].
[19] S. R. Best, ”On the resonant properties of the Koch fractal and other wire monopole antennas,” in IEEE Antennas and Wireless Propagation Letters, vol. 1, pp. 74-76, 2002, doi: 10.1109/LAWP.2002.802550.
[20] P. Huang, D. Li, and H. Zhao, “An Improved Robust Fractal Image Compression Based on M-Estimator,” Applied Sciences, vol. 12, no. 15, p. 7533, Jul. 2022, doi: 10.3390/app12157533.